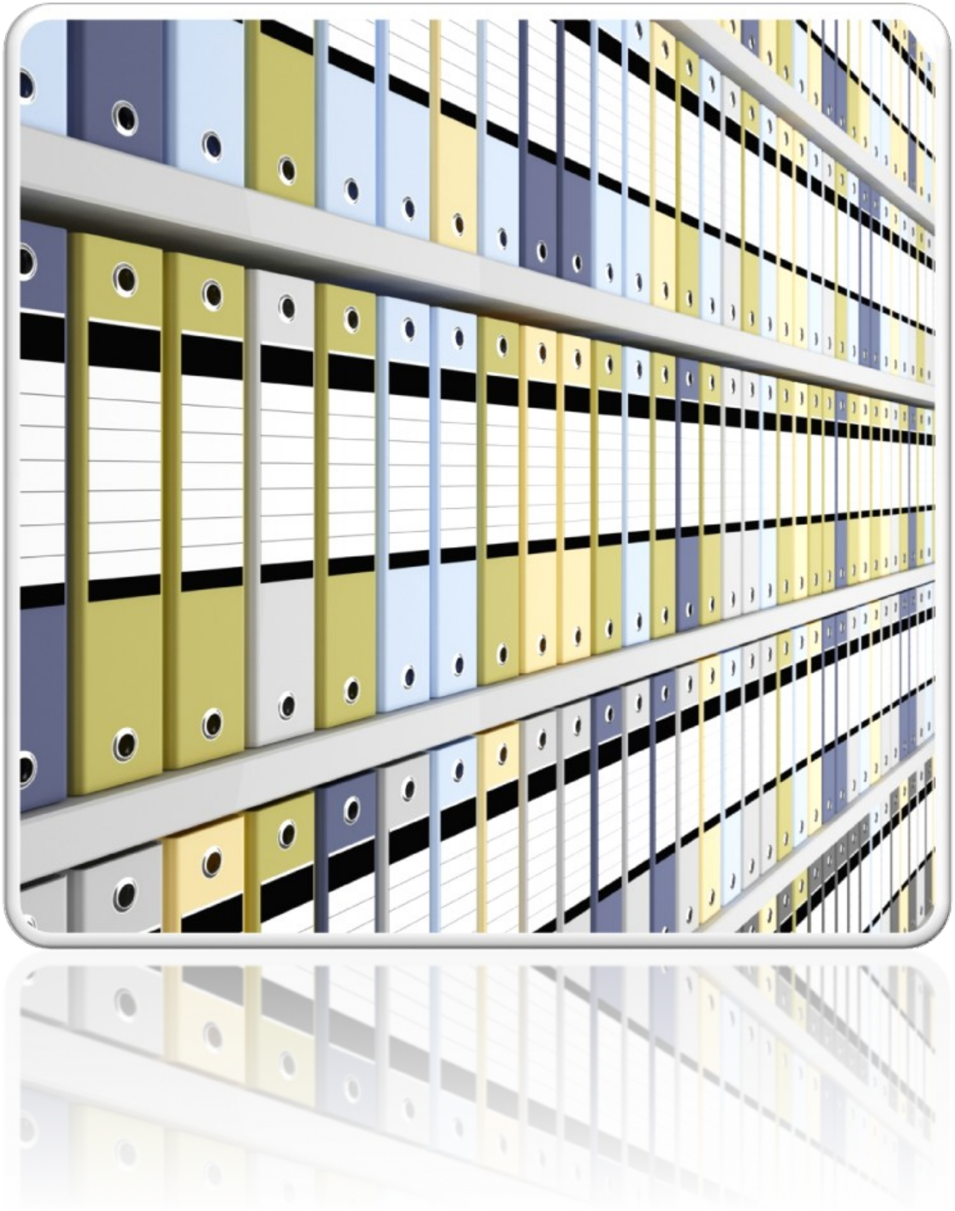
syslog-ng:
log correlation and beyond



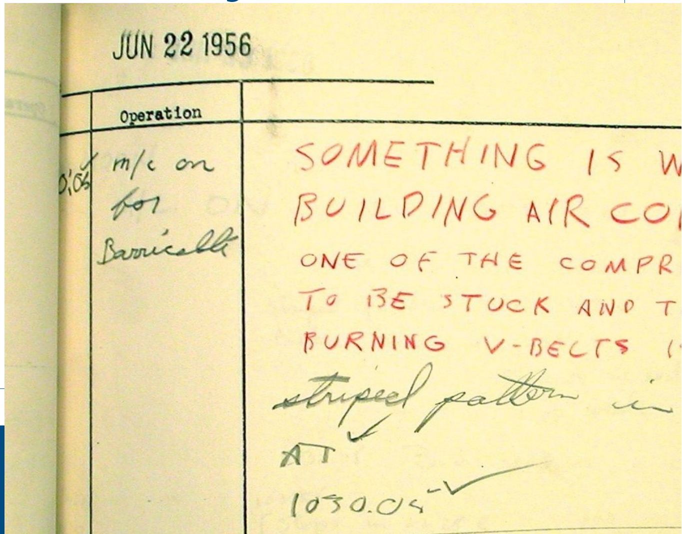
Márton Illés
marton.illes@balabit.com

Contents

- Short introduction to syslog
- The syslog-ng story
- Logging today and SIEMs
- Some new & interesting features in syslog-ng
- Open source SIEM?

Syslog 101

- Spin-off of sendmail by Eric Allmann
- Describing simple events in plain English
- Easy to use API: syslog()
- Messages are stored in files or sent over the network using UDP transport
- Some application simply store messages directly in files, in SQL database or in proprietary format
- Still the most widespread solution
- Only UNIX and network devices



JUN 22 1956

Operation

0:05 m/c on for Barricelli

SOMETHING IS W
BUILDING AIR CO
ONE OF THE COMPR
TO BE STUCK AND T
BURNING V-BELTS (
striped pattern in
AT
1030.05

Problems with the syslog protocol

- No structure at all: hard to parse!
  - Priority and facility is very limited
- Need for central collection, but...
  - No authentication, no encryption, no integrity check, no digital signature
  - No flow-control
  - UDP based transfer with high message loss

```
Jul  3 22:45:21 octane sshd[18206]:
Accepted publickey for marci from 127.0.0.1 port 37126 ssh2
```

The syslog-ng story...

- Designed for central log collection since the beginning
- First release in 1998, now part of most Linux distribution and available for most UNIX flavors
- Operates in multiple global networks serving thousands of devices
- Development funded by BalaBit
  - Open Source Edition, released under GPL
  - Commercial “Premium” and appliance (SSB) editions since 2007/2008

Main features of syslog-ng

- Support for TCP based message transport
  - Understands different syslog flavors (eg: Cisco)
  - Converting between UDP/TCP transports
- Flexible filtering capabilities
- Different, customizable log destinations
  - Message forwarding using TCP
  - File, pipe, program, fifo destinations
  - Utilizing macros and templates
- “Log router” utilizing filters and destinations
- Log parsing and classification using patterndb

Unstructured message parsing

- Parsing unstructured, badly formated messages requires a pattern database
- Most text/message parsing utilizes regular expressions, however...
  - Regexps are hard to write (eg: IPv6 address)
  - Regexps are hard to understand
  - Regexps do not scale to a large number of patterns
  - Regexps do not scale to a high message rate

db-parser()

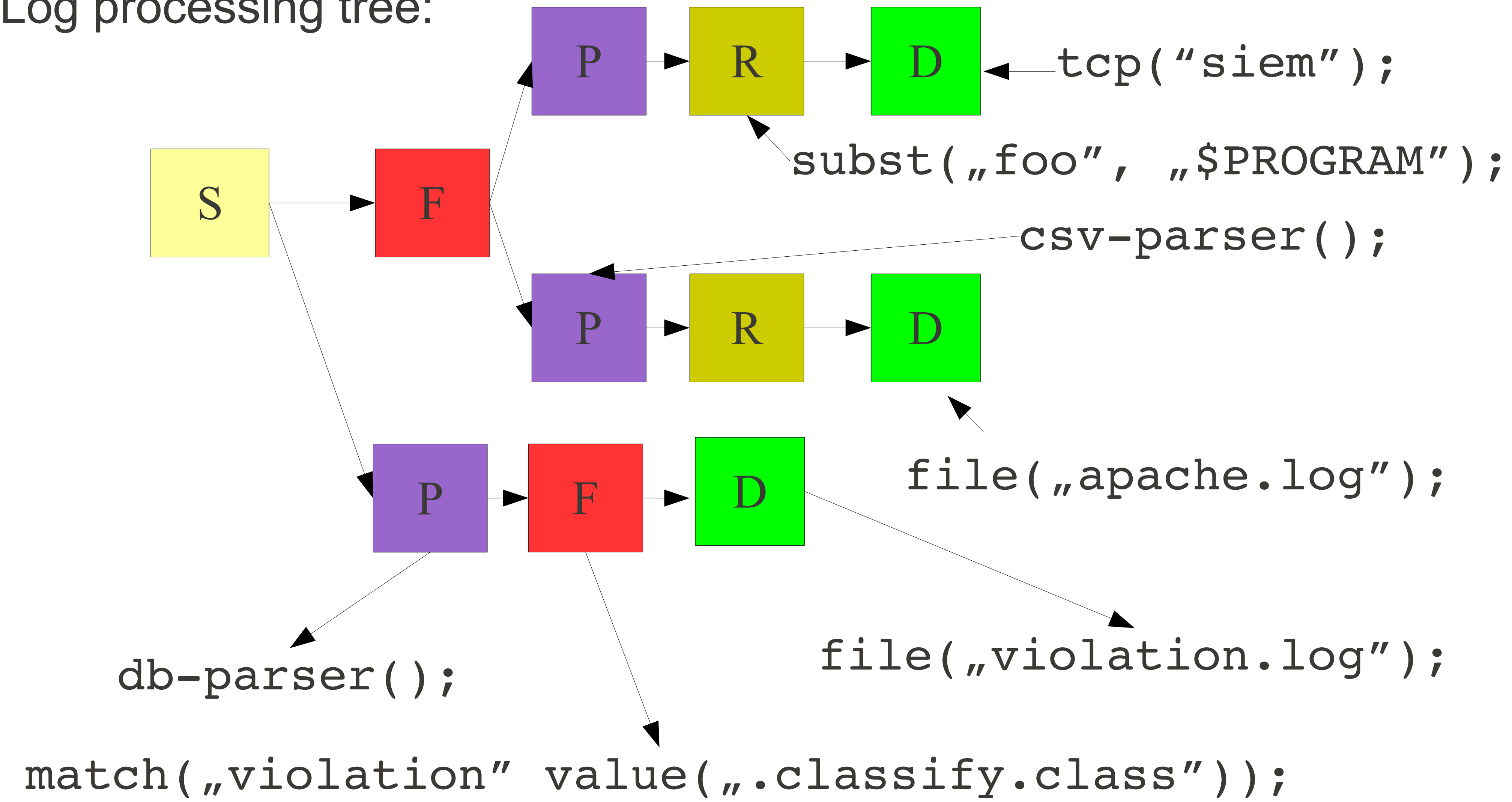
- Syslog-ng parser to parse messages based on a pattern database
  - Recognize, classify, tag messages
  - Extract information from messages
- Performance:
  - Pattern matching costs about 10-20% of performance relative to storing into files
  - Algorithm is close to O(1) on the number of patterns and depends on the length of the msg
- Some pre-defined patterns available as well...

Pattern database example

```
<patterndb version=”2” pub_date=”2009-07-01”>
 <ruleset name=”sshd”>
  <rules>
   <rule id=”1” class=”login”>
    <patterns>
     <pattern>Accepted publickey for @STRING:username@ from
@IPv4:source@ port @NUMBER:port@ ssh2</pattern>
    </patterns>
   </rule>
  </rules>
 </ruleset>
</patterndb>

destination d_sql {
  sql(type(mysql) host(dbhost) database(logs)
table(“login_$R_YEAR_$R_MONTH_$R_DAY) columns(“date
timestamp”, “username”, “source)
values(“$R_UNIXTIME”, “$username”, “$source”));};
```

The “log router”

Log processing tree:



```
tcp(“siem”);
subst(„foo”, „$PROGRAM”);
csv-parser();
file(„apache.log”);
file(„violation.log”);
db-parser();
match(„violation” value(„.classify.class”));
```

Vision of syslog-ng

- Acting as a simple “log router” is not enough anymore
- Syslog-ng needs to aid message analysis
  - Pre-parse message and move them to a common base
  - Extract information from messages
  - Forward messages based on the message content/type/classification
  - NEW: correlate messages and emit aggregates and alerts

New trends in log collection

- Earlier, logs were collected for IT management
  - Troubleshooting, accounting
  - Forensics situations (mainly detective situation)
- The focus and use-cases are changing
  - Security incident and event mgmt. (SIEM)
  - Various regulations
  - Real-time alerting and correlation
  - More messages coming from applications, not just from the infrastructure
- Logs are to be processed automatically

Why correlate messages?

- In some cases a simple event is represented by multiple “independent” messages
  - e.g: postifx, login/logout
- In some cases multiple “independent” event makes a up a real event
  - e.g: port-scans, HTTP requests → sessions
- In some cases a lack of a message/event is the signal of a problem
  - e.g: password failures without successful authentication at the end

What is a SIEM?

- Security Incident and Event Management
- Main operation:
  - Collect events
  - Correlate events
  - Trigger alerts
  - Generate reports, statistics
  - Visualize information not just data
- Real-time and off-line operation
- In many cases they are black-box bloat-wares...

Latest syslog-ng developments - 3.3

- Switch to a module based architecture
- New licensing scheme
  - LGPL core, GLP modules
  - No CLA is required anymore
  - External syslog-ng module repositories
- Multi-threaded operation mode
  - 500,000 messages/sec online processing
- MongoDB (NoSQL) destination
- Template functions (if, echo, grep etc.)

syslog-ng new correlation engine

- Store/lookup states for events as message contexts
  - All matched messages are stored to states
- Trigger new messages based on message states
  - Pre-defined conditions could be used
  - Timeout could be used
  - Rate-limit could be applied
- Part of db_parser() uses patterndb xml database
- Could operate on-line and off-line
  - Work on logfiles using pdbtool

Correlation example

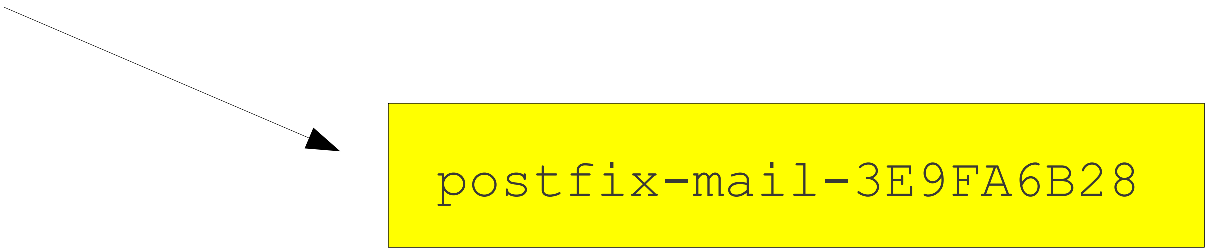
```
 <rule id="123" context-id="postfix-mail-${.postfix.id}" context-timeout="86400"
context-scope="host">
  <patterns>
   <pattern>@ESTRING:.postfix.id::@ from=@QSTRING:.postfix.from:&lt;&gt;@,
size=@ESTRING:.postfix.size:,@</pattern>
  </patterns>
 </rule>

 <rule id="124" context-id="postfix-mail-${.postfix.id}" context-timeout="86400"
context-scope="host">
  <patterns>
   <pattern>@ESTRING:.postfix.id::@ to=@QSTRING:.postfix.to:&lt;&gt;@,
relay=@ESTRING:.postfix.relay:,@ delay=@ESTRING:.postfix.delay:,@
status=@ESTRING:.postfix.status: @</pattern>
  </patterns>
  <actions>
   <action trigger="match">
    <message>
     <values>
      <value name="MSG">Mail accounting;$(grep '${.postfix.from} != ""' $
{.postfix.from});${.postfix.to};${.postifx.status}</value>
     </values>
    </message>
   </action>
  </actions>
 </rule>
```

Correlation example

```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```

Correlation example

```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```



postfix-mail-3E9FA6B28

Correlation example

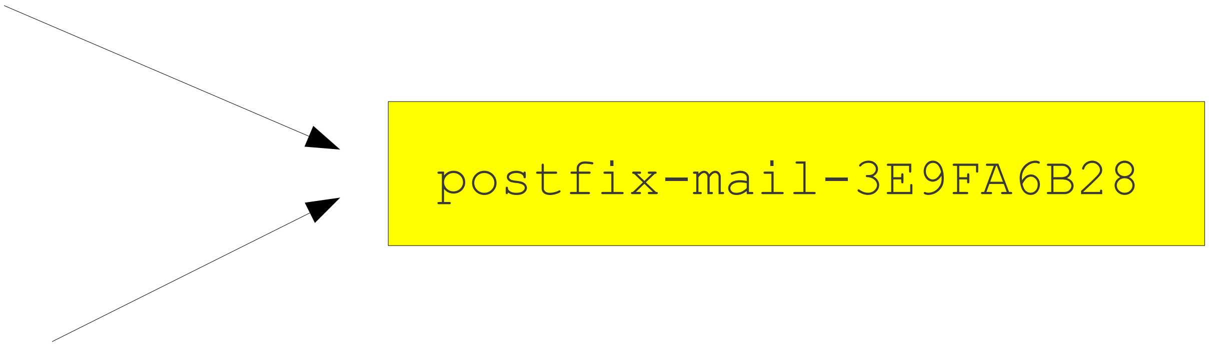
```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```

postfix-mail-3E9FA6B28

```
3E9F4A6B28: to=<rcpt@target.com>, relay=none, delay=0, status=sent
```

Correlation example

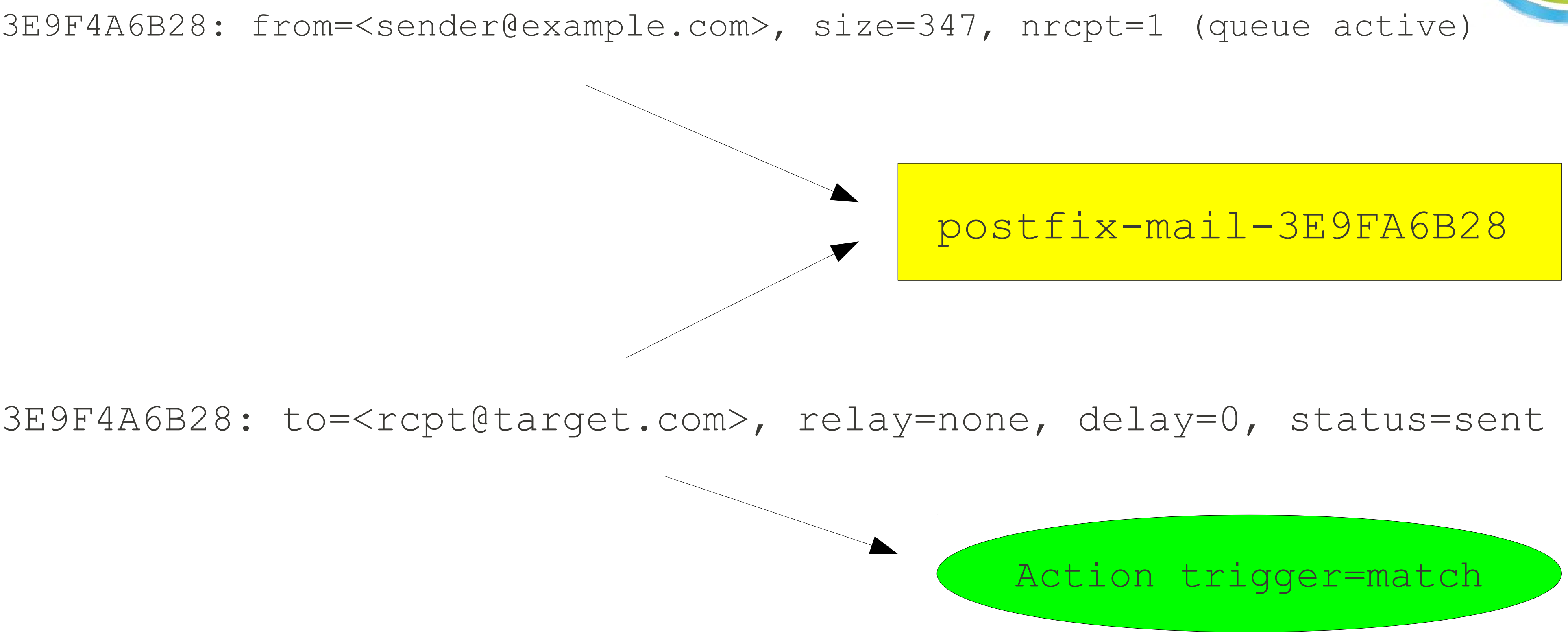
```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```



postfix-mail-3E9FA6B28

```
3E9F4A6B28: to=<rcpt@target.com>, relay=none, delay=0, status=sent
```

Correlation example



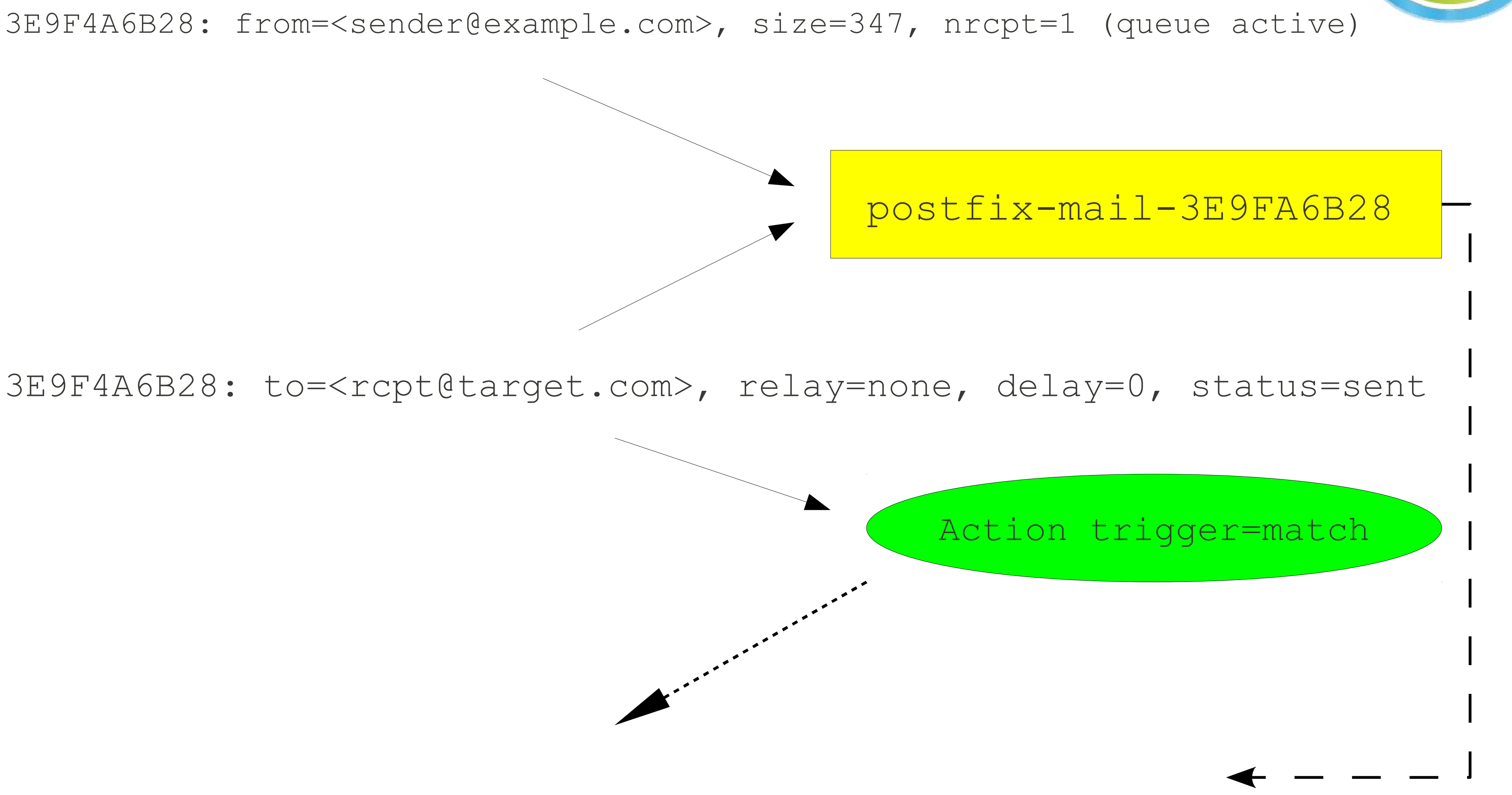
```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```

postfix-mail-3E9FA6B28

```
3E9F4A6B28: to=<rcpt@target.com>, relay=none, delay=0, status=sent
```

Action trigger=match

Correlation example



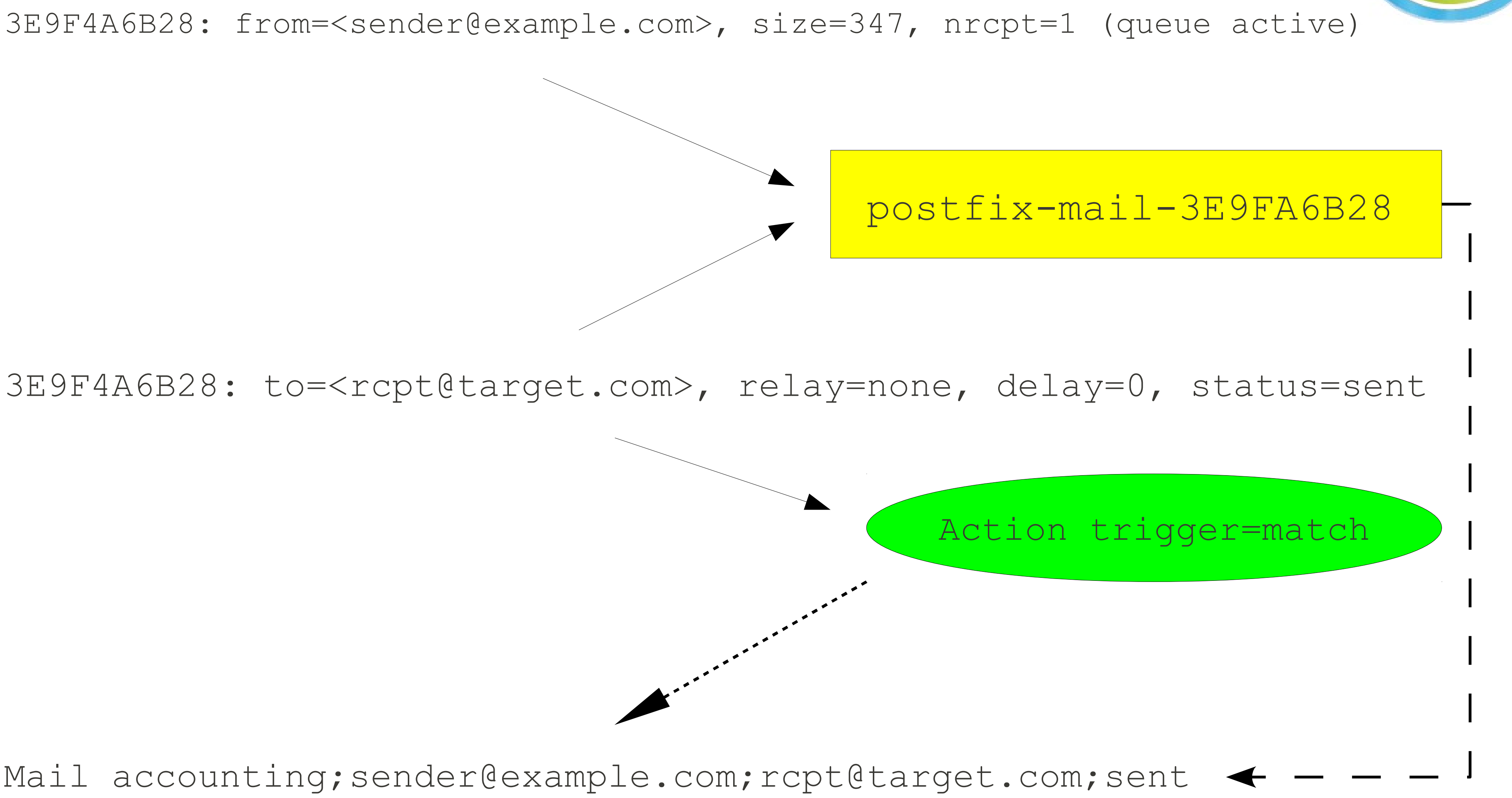
```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```

postfix-mail-3E9FA6B28

```
3E9F4A6B28: to=<rcpt@target.com>, relay=none, delay=0, status=sent
```

Action trigger=match

Correlation example



```
3E9F4A6B28: from=<sender@example.com>, size=347, nrcpt=1 (queue active)
```

postfix-mail-3E9FA6B28

```
3E9F4A6B28: to=<rcpt@target.com>, relay=none, delay=0, status=sent
```

Action trigger=match

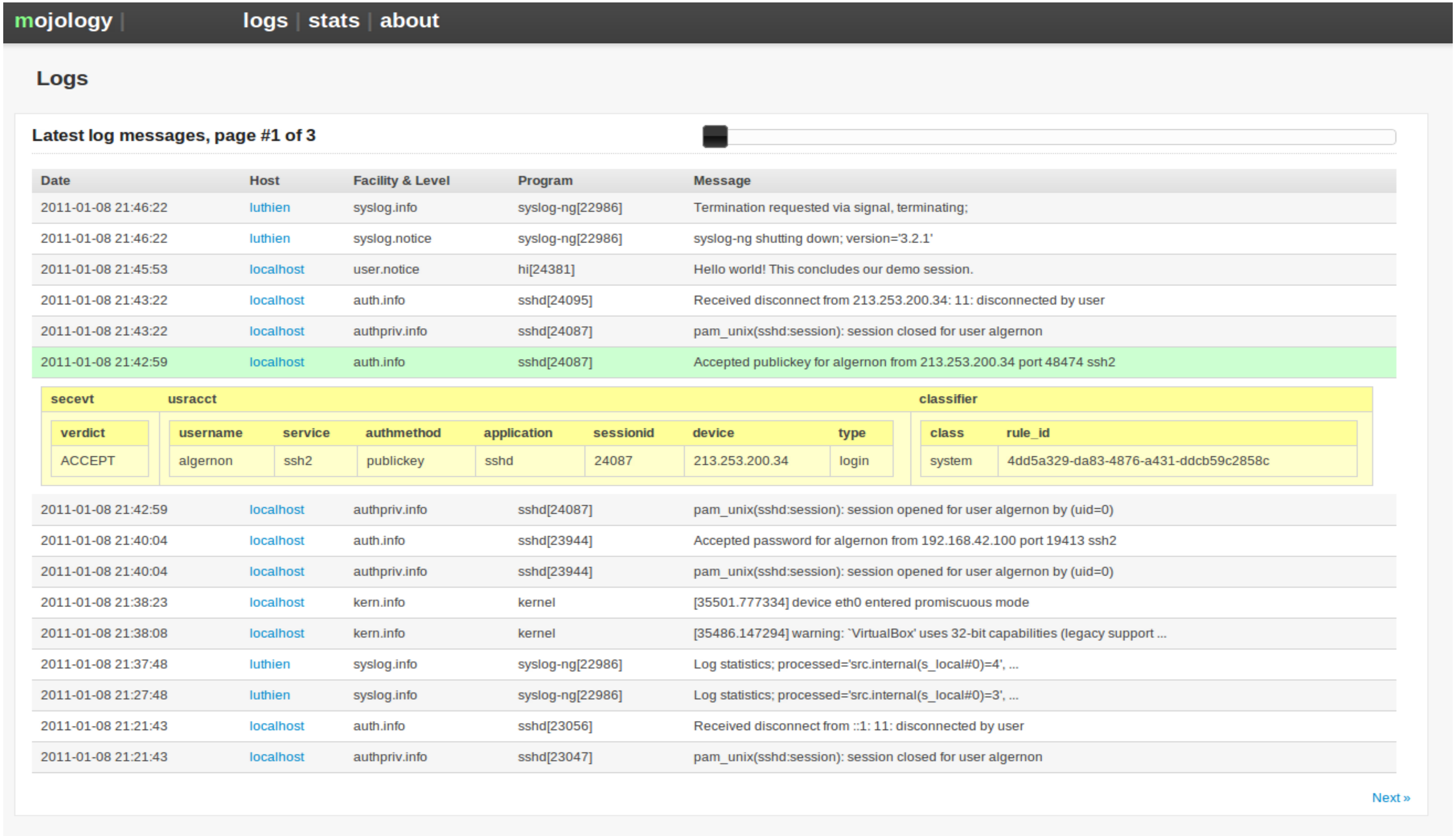
```
Mail accounting;sender@example.com;rcpt@target.com;sent
```

syslog-ng: the base of a simple SIEM?

- Normalize, parse, correlate messages using patterndb rules
- Trigger real-time alerts and send emails, snmp-traps using “program” destination
- Feed existing tools like sec.pl, swatch etc.
- Store results in SQL or in MongoDB
- Generate reports/statistics using simple SQL reporting tools and cron
- Browse, search and visualize logs/events using any SQL frontend or any syslog web interface

Some handy tools I.

- Mojology a MongoDB syslog-ng web front-end



Logs

Latest log messages, page #1 of 3

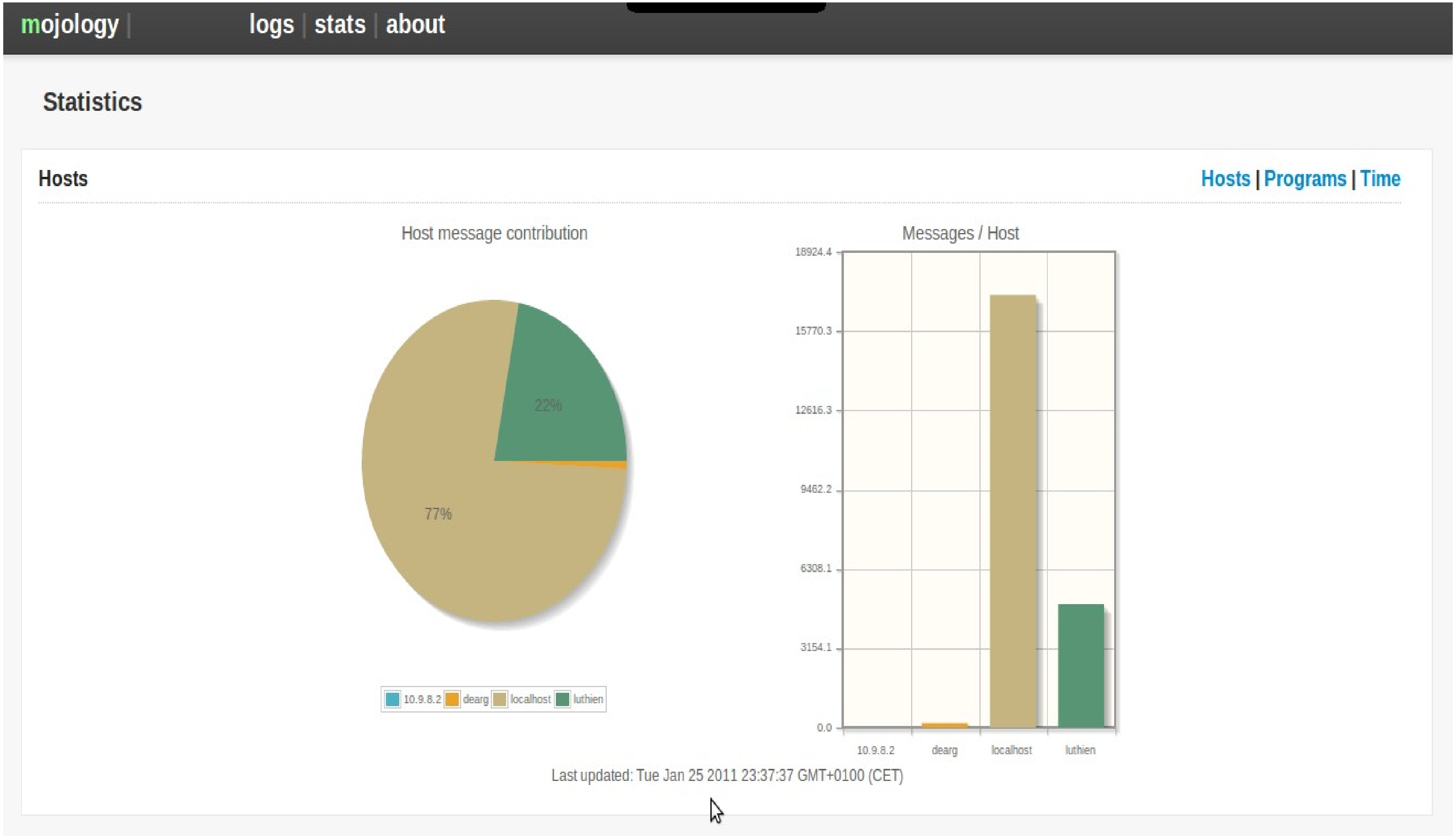
Date	Host	Facility & Level	Program	Message
2011-01-08 21:46:22	luthien	syslog.info	syslog-ng[22986]	Termination requested via signal, terminating;
2011-01-08 21:46:22	luthien	syslog.notice	syslog-ng[22986]	syslog-ng shutting down; version='3.2.1'
2011-01-08 21:45:53	localhost	user.notice	hi[24381]	Hello world! This concludes our demo session.
2011-01-08 21:43:22	localhost	auth.info	sshd[24095]	Received disconnect from 213.253.200.34: 11: disconnected by user
2011-01-08 21:43:22	localhost	authpriv.info	sshd[24087]	pam_unix(sshd:session): session closed for user algernon
2011-01-08 21:42:59	localhost	auth.info	sshd[24087]	Accepted publickey for algernon from 213.253.200.34 port 48474 ssh2

secevt	usracct							classifier	
verdict	username	service	authmethod	application	sessionid	device	type	class	rule_id
ACCEPT	algernon	ssh2	publickey	sshd	24087	213.253.200.34	login	system	4dd5a329-da83-4876-a431-ddcb59c2858c

Date	Host	Facility & Level	Program	Message
2011-01-08 21:42:59	localhost	authpriv.info	sshd[24087]	pam_unix(sshd:session): session opened for user algernon by (uid=0)
2011-01-08 21:40:04	localhost	auth.info	sshd[23944]	Accepted password for algernon from 192.168.42.100 port 19413 ssh2
2011-01-08 21:40:04	localhost	authpriv.info	sshd[23944]	pam_unix(sshd:session): session opened for user algernon by (uid=0)
2011-01-08 21:38:23	localhost	kern.info	kernel	[35501.777334] device eth0 entered promiscuous mode
2011-01-08 21:38:08	localhost	kern.info	kernel	[35486.147294] warning: `VirtualBox' uses 32-bit capabilities (legacy support ...
2011-01-08 21:37:48	luthien	syslog.info	syslog-ng[22986]	Log statistics; processed='src.internal(s_local#0)=4', ...
2011-01-08 21:27:48	luthien	syslog.info	syslog-ng[22986]	Log statistics; processed='src.internal(s_local#0)=3', ...
2011-01-08 21:21:43	localhost	auth.info	sshd[23056]	Received disconnect from ::1: 11: disconnected by user
2011-01-08 21:21:43	localhost	authpriv.info	sshd[23047]	pam_unix(sshd:session): session closed for user algernon

Some handy tools I.

- Mojology a MongoDB syslog-ng web front-end



Statistics

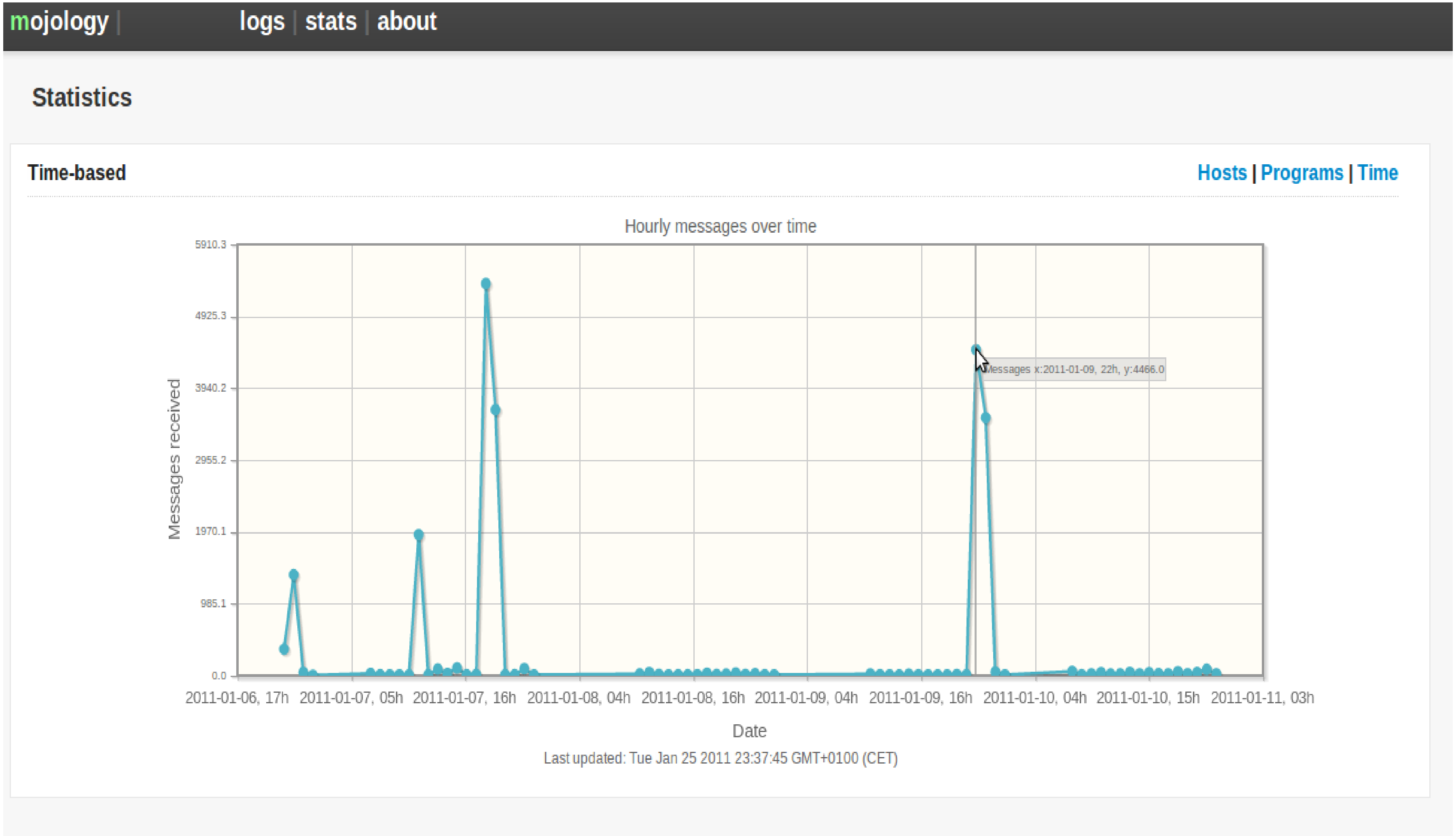
Hosts

Hosts | Programs | Time

Last updated: Tue Jan 25 2011 23:37:37 GMT+0100 (CET)

Some handy tools I.

- Mojology a MongoDB syslog-ng web front-end



mojology | logs | stats | about

Statistics

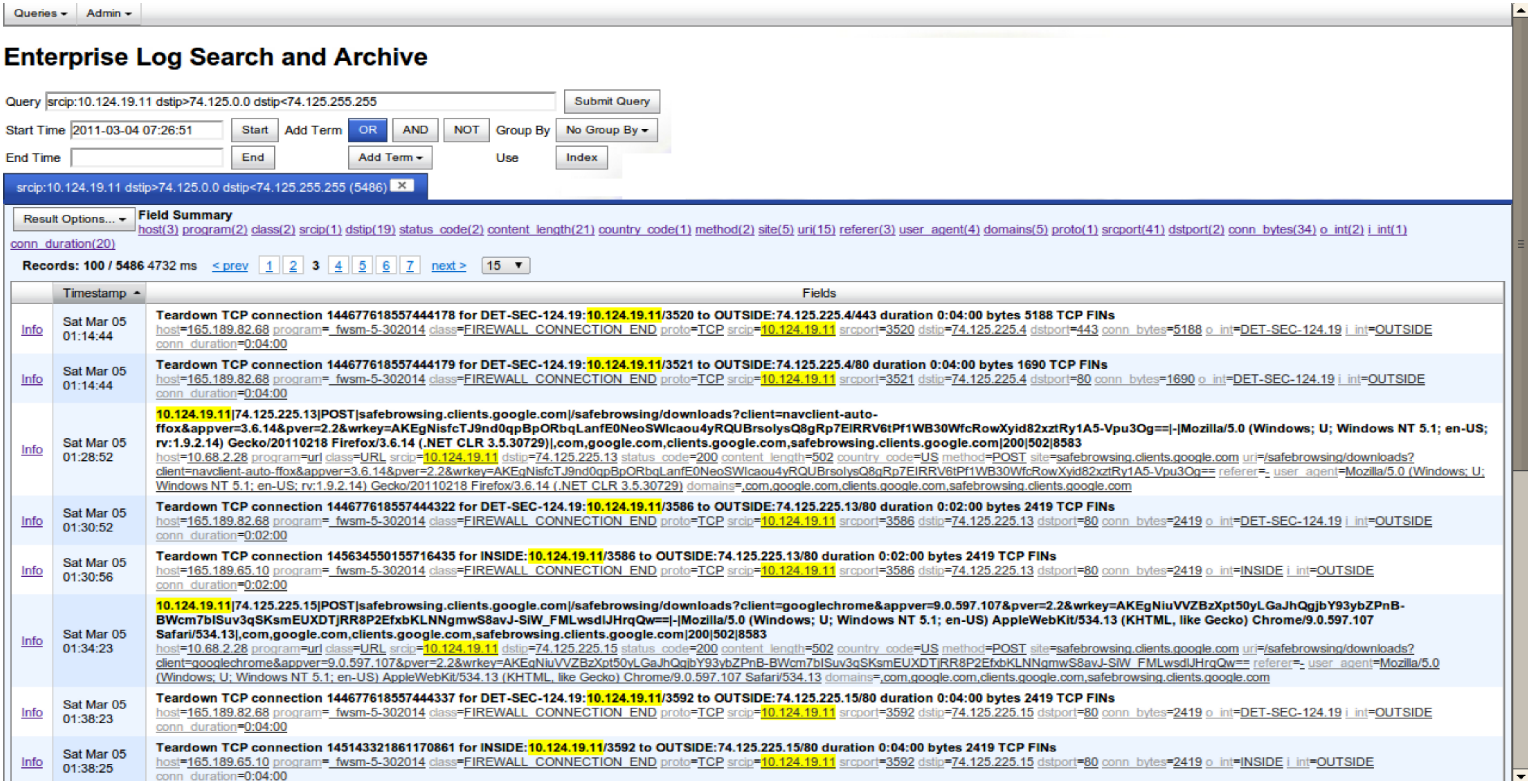
Time-based

Hosts | Programs | Time

Last updated: Tue Jan 25 2011 23:37:45 GMT+0100 (CET)

Some handy tools II.

- ELSA: Enterprise Log Search and Archive
    - syslog-ng, patterndb, sphinx, MySQL



Enterprise Log Search and Archive

Query: srcip:10.124.19.11 dstip>74.125.0.0 dstip<74.125.255.255

Start Time: 2011-03-04 07:26:51

End Time:

End
Add Term ▾
Use
Index
srcip:10.124.19.11 dstip>74.125.0.0 dstip<74.125.255.255 (5486) ✕
Result Options... ▾

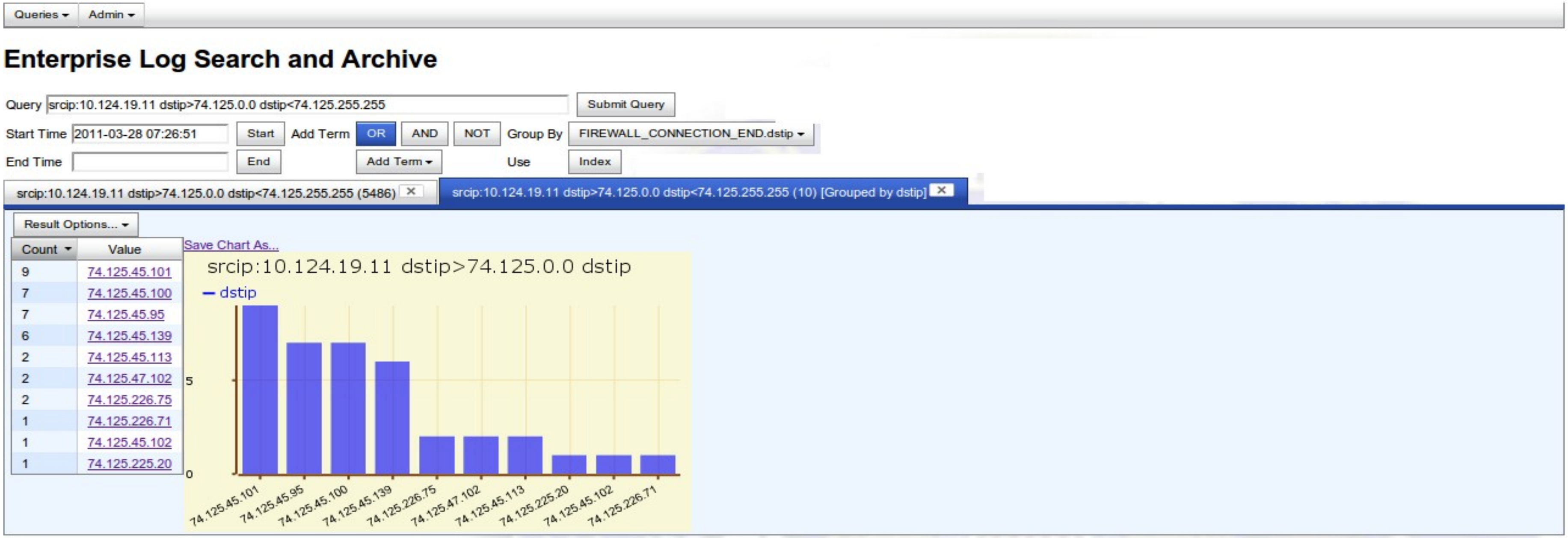
Field Summary
host(3) program(2) class(2) srcip(1) dstip(19) status_code(2) content_length(21) country_code(1) method(2) site(5) uri(15) referer(3) user_agent(4) domains(5) proto(1) srcport(41) dstport(2) conn_bytes(34) o_int(2) i_int(1) conn_duration(20)

Records: 100 / 5486 4732 ms < prev 1 2 **3** 4 5 6 7 next > 15 ▾

```
      Timestamp ▴   Fields

Info  Sat Mar 05    Teardown TCP connection 144677618557444178 for DET-SEC-124.19:10.124.19.11/3520 to OUTSIDE:74.125.225.4/443 duration 0:04:00 bytes 5188 TCP FINs
      01:14:44      host=165.189.82.68 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3520 dstip=74.125.225.4 dstport=443 conn_bytes=5188 o_int=DET-SEC-124.19 i_int=OUTSIDE
                    conn_duration=0:04:00

Info  Sat Mar 05    Teardown TCP connection 144677618557444179 for DET-SEC-124.19:10.124.19.11/3521 to OUTSIDE:74.125.225.4/80 duration 0:04:00 bytes 1690 TCP FINs
      01:14:44      host=165.189.82.68 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3521 dstip=74.125.225.4 dstport=80 conn_bytes=1690 o_int=DET-SEC-124.19 i_int=OUTSIDE
                    conn_duration=0:04:00

Info  Sat Mar 05    10.124.19.11|74.125.225.13|POST|safebrowsing.clients.google.com|/safebrowsing/downloads?client=navclient-auto-
      01:28:52      ffox&appver=3.6.14&pver=2.2&wrkey=AKEgNisfcTJ9nd0qpBpORbqLanfE0NeoSWIcaou4yRQUBrsolysQ8gRp7EIRRV6tPf1WB30WfcRowXyid82xztRy1A5-Vpu3Og==|-|Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
                    rv:1.9.2.14) Gecko/20110218 Firefox/3.6.14 (.NET CLR 3.5.30729)|,com,google.com,clients.google.com,safebrowsing.clients.google.com|200|502|8583
                    host=10.68.2.28 program=url class=URL srcip=10.124.19.11 dstip=74.125.225.13 status_code=200 content_length=502 country_code=US method=POST site=safebrowsing.clients.google.com uri=/safebrowsing/downloads?
                    client=navclient-auto-ffox&appver=3.6.14&pver=2.2&wrkey=AKEgNisfcTJ9nd0qpBpORbqLanfE0NeoSWIcaou4yRQUBrsolysQ8gRp7EIRRV6tPf1WB30WfcRowXyid82xztRy1A5-Vpu3Og== referer=- user_agent=Mozilla/5.0 (Windows; U;
                    Windows NT 5.1; en-US; rv:1.9.2.14) Gecko/20110218 Firefox/3.6.14 (.NET CLR 3.5.30729) domains=,com,google.com,clients.google.com,safebrowsing.clients.google.com

Info  Sat Mar 05    Teardown TCP connection 144677618557444322 for DET-SEC-124.19:10.124.19.11/3586 to OUTSIDE:74.125.225.13/80 duration 0:02:00 bytes 2419 TCP FINs
      01:30:52      host=165.189.82.68 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3586 dstip=74.125.225.13 dstport=80 conn_bytes=2419 o_int=DET-SEC-124.19 i_int=OUTSIDE
                    conn_duration=0:02:00

Info  Sat Mar 05    Teardown TCP connection 145634550155716435 for INSIDE:10.124.19.11/3586 to OUTSIDE:74.125.225.13/80 duration 0:02:00 bytes 2419 TCP FINs
      01:30:56      host=165.189.65.10 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3586 dstip=74.125.225.13 dstport=80 conn_bytes=2419 o_int=INSIDE i_int=OUTSIDE
                    conn_duration=0:02:00

Info  Sat Mar 05    10.124.19.11|74.125.225.15|POST|safebrowsing.clients.google.com|/safebrowsing/downloads?client=googlechrome&appver=9.0.597.107&pver=2.2&wrkey=AKEgNiuVVZBzXpt50yLGaJhQgjbY93ybZPnB-
      01:34:23      BWcm7blSuv3qSKsmEUXDTjRR8P2EfxbKLNNgmwS8avJ-SiW_FMLwsdIJHrqQw==|-|Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.107
                    Safari/534.13|,com,google.com,clients.google.com,safebrowsing.clients.google.com|200|502|8583
                    host=10.68.2.28 program=url class=URL srcip=10.124.19.11 dstip=74.125.225.15 status_code=200 content_length=502 country_code=US method=POST site=safebrowsing.clients.google.com uri=/safebrowsing/downloads?
                    client=googlechrome&appver=9.0.597.107&pver=2.2&wrkey=AKEgNiuVVZBzXpt50yLGaJhQgjbY93ybZPnB-BWcm7blSuv3qSKsmEUXDTjRR8P2EfxbKLNNgmwS8avJ-SiW_FMLwsdIJHrqQw== referer=- user_agent=Mozilla/5.0
                    (Windows; U; Windows NT 5.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.107 Safari/534.13 domains=,com,google.com,clients.google.com,safebrowsing.clients.google.com

Info  Sat Mar 05    Teardown TCP connection 144677618557444337 for DET-SEC-124.19:10.124.19.11/3592 to OUTSIDE:74.125.225.15/80 duration 0:04:00 bytes 2419 TCP FINs
      01:38:23      host=165.189.82.68 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3592 dstip=74.125.225.15 dstport=80 conn_bytes=2419 o_int=DET-SEC-124.19 i_int=OUTSIDE
                    conn_duration=0:04:00

Info  Sat Mar 05    Teardown TCP connection 145143321861170861 for INSIDE:10.124.19.11/3592 to OUTSIDE:74.125.225.15/80 duration 0:04:00 bytes 2419 TCP FINs
      01:38:25      host=165.189.65.10 program=_fwsm-5-302014 class=FIREWALL_CONNECTION_END proto=TCP srcip=10.124.19.11 srcport=3592 dstip=74.125.225.15 dstport=80 conn_bytes=2419 o_int=INSIDE i_int=OUTSIDE
                    conn_duration=0:04:00
```

Some handy tools II.

- ELSA: Enterprise Log Search and Archive
  - syslog-ng, patterndb, sphinx, MySQL



Enterprise Log Search and Archive

Query: srcip:10.124.19.11 dstip>74.125.0.0 dstip<74.125.255.255

Start Time: 2011-03-28 07:26:51

Group By: FIREWALL_CONNECTION_END.dstip ▾

End Time:

srcip:10.124.19.11 dstip>74.125.0.0 dstip<74.125.255.255 (5486) ✕

srcip:10.124.19.11 dstip>74.125.0.0 dstip<74.125.255.255 (10) [Grouped by dstip] ✕

Count ▾	Value
9	74.125.45.101
7	74.125.45.100
7	74.125.45.95
6	74.125.45.139
2	74.125.45.113
2	74.125.47.102
2	74.125.226.75
1	74.125.226.71
1	74.125.45.102
1	74.125.225.20

A simple solution I liked very much :)

- Securing servers with iptables against ssh brute-force attacks using syslog-ng db_parser
- Use iptables recent match to block addresses
- Use patterndb to detect SSH auth failures and to extract “attackers” source IP address
- Use custom syslog-ng destination file template to feed “recent” match's database
- Idea and example by Valentijn Sessink

A simple solution I liked very much :)

```
# an iptables-destination in /proc to block addresses
destination d_syslogblock {
   file("/proc/net/xt_recent/syslogblock"
      template("+${usracct.device}\n"));
};

# a parser for the pattern-DB we made
parser pattern_db {
    db_parser( file("/var/lib/syslog-ng/patterndb.xml")); };

# a filter to filter the parser results
filter f_syslogblock {
   tags("secevt") and match("REJECT"
      value("secevt.verdict"));
};

# and finally, the log itself:
log {
   source(s_src); parser(pattern_db); filter(f_syslogblock);
destination(d_syslogblock);
};
```

Summary

- There are severe problems how logging is done today
- More logs are coming from more applications
- Logs need to be processed not just stored
- Many problems could be solved with simple open-source tools without complex and expensive SIEMs
- syslog-ng could help not just with message collection, but also with message processing

Some useful links

- http://algernon.blogs.balabit.com/2011/03/the-birth-of-mojology/
- http://valentijn.sessink.nl/?p=322
- http://ossectools.blogspot.com/2011/03/fighting-apt-with-open-source-software.html
- http://lwn.net/Articles/424459/

Questions and answers

Merci de votre attention!

Márton Illés
marton.illes@balabit.com